



UNIVERZITET U NOVOM SADU
Fakultet tehnickih nauka u Novom Sadu

Distribuirani računarski sistemi u EE

-Primenjeno softversko inženjerstvo –

Projekat „Turn Snake“

Tim „Fatal error“

Milan Momčilović PR 33/2017

Milica Simeunović PR 9/2017

Katarina Veljković PR 12/2017

Aleksandra Gaicki PR 105/2017

Novi Sad, 30.1.2021

Opis aplikacije

Aplikacija “Turn Snake” je strateška igra bazirana na potezima koja se igra od 2 do 4 igrača. Svaki igrač poseduje jednu ili više zmija kojima može da se kreće po terenu sa ciljem da što duže opstane u igri, tako što će sakupljati hranu i pokušavati da zarobi ostale igrače.

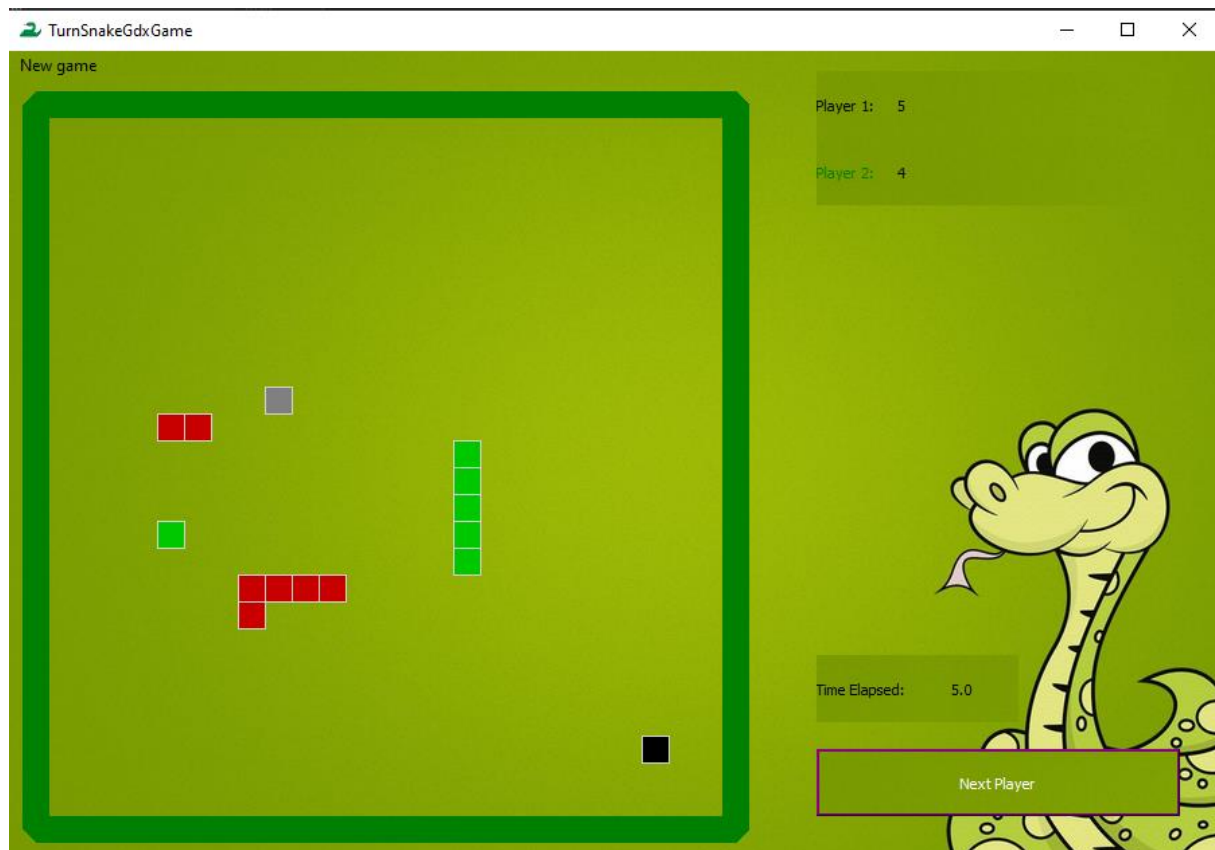
Mehanika igre

Na samom početku igre korisnici mogu odabrati broj igrača, broj zmija po igraču i vremensko ograničenje svakog poteza. Igrač koji je na redu ima određeni vremenski interval u kom može da odigra svoj potez, tj. pomeri zmije maksimalan broj koraka u izabranom smeru. Ukoliko završi potez, ili je vreme isteklo, aplikacija automatski prelazi na sledećeg igrača, a igrača je moguće promeniti i klikom na dugme “Next player” ili pritiskom na taster “Enter”. Svaki igrač može u toku svog poteza da menja zmije sa kojima će se pomerati, pritiskom na taster “N”.

Posle svakog odigranog koraka proverava se da li je došlo do kolizije između zmija, hrane, zida terena kao i da li je zmija opkoljena:

- **Zid:** Ukoliko zmija udari u zid, ona nestaje sa terena, a ukoliko je to jedina zmija koju je igrač posedovao, taj igrač završava igru
- **Hrana:** Zmija može da pojede hranu i time povećava dužinu svog tela i broj koraka za 1. Hrana se posle svakog poteza pomera pravolinijski nekoliko koraka u proizvoljnom smeru.
- **Specijalna hrana:** Ova hrana se pojavljuje na određenoj poziciji na terenu nakon određenog vremenskog intervala i ukoliko je zmija pojede može se desiti da igrač dobije još 3 koraka po potezu i poveća svoj rezultat ili da izgubi 1 korak i smanji broj poena.
- **Kolizija između zmija:** Ukoliko glava zmije trenutnog igrača udari u telo druge zmije ili u samu sebe ona nestaje iz igre.
- **Opkoljavanje:** Ukoliko je neka zmija opkoljena od strane protivničke zmije, tj. nije u mogućnosti da se pomera na bilo koju stranu, ta zmija nestaje sa terena.

Igra se završava onda kada je ostao samo jedan igrač na terenu. Smatra se da je igrač izgubio onog trenutka kada nestanu sve njegove zmije sa terena. Poslednji aktivni igrač se smatra pobednikom.



Slika 1. Izgled ekrana igre

Softverska implementacija

Opšta arhitektura softverske implementacije je kombinacija više-procesne arhitekture i višeslojne arhitekture

Sa više-procesnog pogleda na arhitekturu primećujemo sledeće komponente:

- **GameApplication:** sadrži stanje igre, upravlja mehanikom igre, prima komande od **Example**-a koje izvršava i obavještava **Example** o novom stanju unutar igre.
- **Example:** Prikazuje korisniku trenutno stanje igre kroz grafičku predstavu. Prima od klijenta komande koje prosleđuje **GameApplication**-u. Od **GameApplication** prima obaveštenja o novom stanju igre koje prikazuje korisniku.

Sa višeslojnog arhitekturnog pogleda primećujemo sledeće slojeve:

- **Sloj mehanike igre:** Sadrži entitete koji učestvuju u igri i mehaniku ponašanja tih entiteta koji grade pravila igre.
- **Sloj komunikacije između procesa:** Sadrži pipe-ove preko kojih procesi komuniciraju i razmenjuju podatke.
- **Sloj prezentacije:** Sadrži grafiku koja predstavlja entitete iz mehanike igre. Sadrži interfejs kojim se neposredno upravlja entitetima igre.

Prednosti i mane Python jezika

Python je programski jezik visokog nivoa opšte namene. Podržava imperativni, objektno-orijentisan i funkcionalni stil programiranja.

Prednosti:

- Sintaksa jezika Pajton omogućava pisanje veoma preglednih programa, zbog čega se jezik veoma brzo i lako uči.
- Postoje različite implementacije interpretera bazirane na već postojećim virtuelnim mašinama, pa je zbog toga npr. postojeći Java kod lako integrisati u Pajton.
- Pisanje malih programa je veoma brzo zato što nije striktno tipski orijentisan i jer posjeduje veliku standardnu biblioteku koja dolazi uz interpreter.
- Mogućnost ubacivanje podataka u rečnik, koji su različitog tipa.

Stil pisanja koda je sastavni dio Pajton sintakse pa su programi pisani u Pajtonu uvijek strukturno uredni.

Mane:

- Rad na velikim projektima i u velikim grupama može biti veoma otežano zato što Pajton nije striktno tipski orijentisan.
- IDE alati teže mogu da predvide i ponude željeno automatsko dopunjavanje koda zato što je tip promenljivih često u potpunosti nepoznat.
- Postojanje ugrađenih metoda koje su pisane camel case-om, iako je standard Pajtona da se pišu sa crticama.

Prednosti i mane PyQt5 okvira

Qt je opšti višepatformski aplikacioni interfejs, koji se koristi za razvoj grafički orijentisanih programa, ali i za razvoj negrafičkih programa, poput konzolnih servera.

Prva verzija se pojavila 1995. godine, a do dan danas Qt se aktivno razvija. Pisan u C++ jeziku. PyQt5 Pajton biblioteka koja vrapuje Qt interfejs pisan u C++ jeziku da bi se mogao lako koristiti u Pajton programima.

Prednosti PyQt5 okvira:

- Višepatformska biblioteka koja korisnika biblioteke sakriva od detalja konkretne implementacije funkcija koje su usko vezane za operativni sistem kada je u pitanju crtanje grafike.
- Omogućuje jednostavno pisanje kompleksnih grafičkih programa.
- Podržava specijalizovane threading biblioteke koje olakšavaju paralelni rad u grafičkim aplikacijama.

Mane PyQt5 okvira:

- Ne postoji dizajner koji olakšava izradu rasporeda grafičkih komponenti.
- Raspored komponenti je definisan i usko vezan uz izvorni kod.
- Nije biblioteka za pravljenje igrica.
- Loša dokumentacija.

Paralelizacija rada

Pajton programski jezik u standardnoj biblioteci nudi podršku za više-procesnu i više-nitnu paralelizaciju rada.

Takođe nudi sve potrebne klase koje omogućavaju sinhronizaciju i komunikaciju između niti i/ili procesa. Veoma je bitno pri paralelizaciji rada biti svjestan da više-nitna paralelizacija u CPython implementaciji dele jednu sistemsku nit tako da se ne može uposliti više procesa ukoliko ih sistem poseduje.

Za pravu paralelizaciju potrebno je koristiti višeprocesnu paralelizaciju.

Zaključak

Zbog pozitivnih osobina koje su navedene u ovom dokumentu, ukoliko je potrebno napisati manju aplikaciju Pajton je odličan izbor. Velika prednost jesu njegove velike biblioteke i popularnosti koju posjeduje.

Stroga netipiziranost jezika je predstavljala problem u komunikaciji tima jer je često dovodilo do potrebe da se dodatno objašnjavaju parametri i povratne vrednosti funkcija.

PyQt5 okvir je bio potpuno pogrešan izbor za izradu grafičkog dela igre i veoma je otežao proces grafičkog razvoja i nije mogao dati dobre performanse jer nije postojala mogućnost direktnog pristupa grafičkom hardveru.